Benchmark FAQ

Q1. What is the RFP performance metric?

A1. Proposed system throughput as measured by the RFP throughput benchmark is the basis for the performance metric evaluation. The baseline throughput performance will be measured in a 6 hour window of the throughput workstreams on their current target IT architectures. PE counts for each workstream instance will be adjusted to fill the current target IT architecture for the 6 hour test window. The number of simulation time units achieved in that 6 hour window is the performance baseline for each workstream. Tentative workstream throughput widths (i.e. the number of instances of a workstream) are provided in the section J draft. Work is ongoing to make the baseline throughput performance available to Offerors as quickly as possible. Offerors will propose the number of throughput suites for a given workstream (i.e. the System Life Throughput) to be supplied over the term of the contract as described in Section C.6.1.2 as the metric of performance to be evaluated for the RFP response.

Q2. If the RFP performance metric is the throughput benchmark, what is the purpose of the workstream component scaling studies?

A2. The scaling study is not used in and of itself as a performance metric. Rather, the scaling study provides information into two aspects of the proposed system. First, the scaling information is an essential element in evaluating the risk of projections used to determine the offered system performance. Second, the scaling study provides information concerning the “performance point” for a given application on the offered system (i.e. Where in the performance curve is a workstream component being offered on the proposed target IT architecture? Is there any performance “headroom” left?).

Q3. How does one verify that a model is running “correctly”?

A3. As is well unknown in the Weather and Climate communities, the systems being modeled are highly non-linear. As a consequence, even small differences due to order of operation changes quickly propagate to produce significant differences in many program variables. Platform differences in mathematical libraries only compound the problem. Therefore only by making long runs and demonstrating that the results for model global values are statistically “the same” can one draw the conclusion that the models are getting the same answer across platforms.

Unfortunately this approach is completely impractical in the context of a benchmark. Thus the Government is considering the following approach:

1. “Basic verification”: No models produce floating point (FP) errors (such as NaNs) on their current platforms. Production of FP errors on vendor platforms is indicative of a problem. Such problems are perhaps due to coding errors already

present in the model or some issues with the vendor platform. The source of FP errors must be identified and rectified.

2. “Self consistency”: Many of the models will reproduce across processor counts at least for certain settings of input parameters. The models in the category are WS 1 and 2 (the CM2 models), WS5 (GFS), WS 7, 8 and 9 (RUC and both examples of this WRF model). Moreover, global sums should be within order of operation differences regardless of reproducibility settings. Reproducibility tests for each of these models are being specified as part of the scaling study. See the latest draft release of the Section J, section J.1.4.3.2, “Running the RDHPCS Scaling Study” for the current statement. NOTE: Statements for reproducibility tests for those codes which do not bitwise reproduce across processor count are still in the process of being developed.

3. “Short run verification of particular model variables”: By the time of RFP release, an explicit set of verification values will be provided for all benchmark components. Example verification files will be provided via the rdhpcs.noaa.gov website. All verification will be based on the ascii output specified for return with the benchmark results. Explicit variables will be specified for verification. Verification will be based on “short” runs in order to minimize difference propagation. It is currently anticipated that tolerance for these verification values will be in the range of 10-20% (i.e. vendor values should be within 10-20% of the verification value).

Q4. How far may code be changed to accommodate new parallel communication methods?

A4. Changes to code in the subdirectories of shared/ (WS1-3) or SMS (WS7) or RSL (WS4, 8 and 9) would be viewed as class B changes. There is no hierarchy implied by the letter designations... "class B" simply means a change to "communication". For example, a reimplementation of the exchange grid in WS 1&2 (shared/exchange/xgrid.F90) would simply be class B. Similarly, use of compiler directives is clearly class C. Again, there's no negative connotation to a class C change. Explicit restructuring of code is problematic to generalize. It is the case that we must review class C changes with the code author. But we've found the science community very flexible when substantial performance enhancement can be demonstrated. Even more careful consideration must be given when a proposed code restructuring may have a negative performance impact on other platforms. Important questions such as:

- Can the change be “isolated” from other platforms (such as through pre-processor directives)?
- How pervasive are the changes?
- Are the code authors willing to support the change over the useful life of the code?

Introduction of non-ANSI standard syntax "above" the shared infrastructure is problematic and comes under the heading of class D. Class D changes are not grounds for immediate "disqualification". But acceptance of such changes would be highly workstream dependent (i.e., various research groups represented by the different workstreams are likely to take very different views on how acceptable class D changes are). Moreover, the performance gain would have to be tremendous and well documented. Additionally, such changes would have to be "maskable" from other architectures (as through preprocessing). The best chance for getting such class D changes accepted would be:

- Document compelling performance improvement from the changes.
- Keep such changes highly "local" both in number of files affected and the number of lines changed within the file.

Make sure you've presented complete performance numbers without the changes and use an incremental approach if necessary (i.e. submit multiple sets of performance numbers). As the RFP states: 'a performance value and the set of associated changes will be evaluated as a single entity and accepted or rejected as such' (i.e., we cannot guess what the performance might have been if only an unacceptable change were removed).